**What is a web server?**

In this article we go over what web servers are, how they work, and why they're important.

|  |  |
| --- | --- |
| **Prerequisites:** | You should already know [how the Internet works](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work), and [understand the difference between a web page, a web site, a web server, and a search engine](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Pages_sites_servers_and_search_engines). |
| **Objective:** | You will learn what a web server is and gain a general understanding of how it works. |

**Summary**

**"Web server"** can refer to hardware or software, or both of them working together.

1. On the hardware side, a web server is a computer that stores web server software and a website's component files (e.g. HTML documents, images, CSS stylesheets, and JavaScript files). It is connected to the Internet and supports physical data interchange with other devices connected to the web.
2. On the software side, a web server includes several parts that control how web users access hosted files, at minimum an *HTTP server*. An HTTP server is a piece of software that understands [URLs](https://developer.mozilla.org/en-US/docs/Glossary/URL) (web addresses) and [HTTP](https://developer.mozilla.org/en-US/docs/Glossary/HTTP) (the protocol your browser uses to view webpages). It can be accessed through the domain names (like mozilla.org) of websites it stores, and delivers their content to the end-user's device.

At the most basic level, whenever a browser needs a file which is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct web server (hardware), the *HTTP server* (software) accepts request, finds the requested document (if it doesn't then a [404](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404) response is returned), and sends it back to the browser, also through HTTP.

To publish a website, you need either a static or a dynamic web server.

A **static web server**, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files "as-is" to your browser.

A **dynamic web server** consists of a static web server plus extra software, most commonly an *application server* and a *database*. We call it "dynamic" because the application server updates the hosted files before sending them to your browser via the HTTP server.

For example, to produce the final webpages you see in the browser, the application server might fill an HTML template with contents from a database. Sites like MDN or Wikipedia have many thousands of webpages, but they aren't real HTML documents, only a few HTML templates and a giant database. This setup makes it easier and quicker to maintain and deliver the content.

**Deeper dive**

To fetch a webpage, as we already said, your browser sends a request to the web server, which proceeds to search for the requested file in its own storage space. On finding the file, the server reads it, processes it as needed, and sends it to the browser. Let's look at those steps in more detail.

Hosting files

A web server first has to store the website's files, namely all HTML documents and their related assets, including images, CSS stylesheets, JavaScript files, fonts, and videos.

Technically, you could host all those files on your own computer, but it's far more convenient to store them all on a dedicated web server that

* is always up and running
* is always connected to the Internet
* has the same IP address all the time (not all [ISPs](https://developer.mozilla.org/en-US/docs/Glossary/ISP) provide a fixed IP address for home lines)
* is maintained by a third-party provider

For all these reasons, finding a good hosting provider is a key part of building your website. Dig through the various services companies offer and choose one that fits your needs and your budget (services range from free to thousands of dollars per month). You can find more details [in this article](https://developer.mozilla.org/en-US/Learn/How_much_does_it_cost#Hosting).

Once you set up a web hosting solution, you just have to [upload your files to your web server](https://developer.mozilla.org/en-US/docs/Learn/Upload_files_to_a_web_server).

Communicating through HTTP

Second, a web server provides support for [HTTP](https://developer.mozilla.org/en-US/docs/Glossary/HTTP) (**H**yper**t**ext **T**ransfer **P**rotocol). As its name implies, HTTP specifies how to transfer hypertext (i.e., linked web documents) between two computers.

A [Protocol](https://developer.mozilla.org/en-US/docs/Glossary/Protocol) is a set of rules for communication between two computers. HTTP is a textual, stateless protocol.

**Textual**

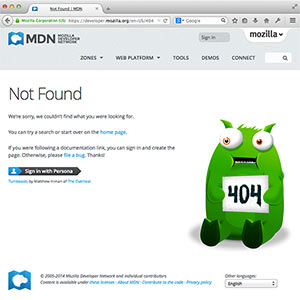
All commands are plain-text and human-readable.

**Stateless**

Neither the server nor the client remember previous communications. For example, relying on HTTP alone, a server cannot remember a password you typed or what step you're on in a transaction. You need an application server for tasks like that. (We'll cover that sort of technology in further articles.)

HTTP provides clear rules for how a client and server communicate. We'll cover HTTP itself in a [technical article](https://developer.mozilla.org/en-US/docs/Web/HTTP) later on. For now, just be aware of these things:

* Only *clients* can make HTTP requests, and then only to *servers*. Servers can only *respond* to a *client*'s HTTP request.
* When requesting a file via HTTP, clients must provide the file's [URL](https://developer.mozilla.org/en-US/docs/Glossary/URL).
* The web server *must answer*every HTTP request, at least with an error message.

[](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404)On a web server, the HTTP server is responsible for processing and answering incoming requests.

1. On receiving a request, an HTTP server first checks whether the requested URL matches an existing file.
2. If so, the web server sends the file content back to the browser. If not, an application server builds the necessary file.
3. If neither process is possible, the web server returns an error message to the browser, most commonly [404 Not Found](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404). (That error is so common that many web designers spend quite some time designing [404 error pages](http://www.404notfound.fr/).)

Static vs. dynamic content

Roughly speaking, a server can serve either static or dynamic content. "Static" means "served as-is". Static websites are the easiest to set up, so we suggest you make your first site a static site.

"Dynamic" means that the server processes the content or even generates it on the fly from a database. This solution provides more flexibility, but the technical stack becomes more difficult to handle, making it dramatically more complex to build the website.

Take for example the page you're reading right now. On the web server hosting it, there is an application server that takes article content from a database, formats it, puts it inside some HTML templates, and sends you the results. In this case, the application server is called [Kuma](https://developer.mozilla.org/en-US/docs/MDN/Kuma) and is built with [Python](https://www.python.org/) (using the [Django](https://www.djangoproject.com/) framework). The Mozilla team built Kuma for the specific needs of MDN, but there are many similar applications built on many other technologies.

There are so many application servers that it's pretty hard to suggest a particular one. Some application servers cater to specific website categories like blogs, wikis or e-shops; others, called [CMSs](https://developer.mozilla.org/en-US/docs/Glossary/CMS) (content management systems), are more generic. If you're building a dynamic website, take the time to choose a tool that fits your needs. Unless you want to learn some web server programming (which is an exciting area in itself!), you don't need to create your own application server. That's just [reinventing the wheel](https://en.wikipedia.org/wiki/reinventing%20the%20wheel).

**Next steps**

Now that you are familiar with web servers, you could:

* read up on [how much it costs to do something on the web](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_much_does_it_cost)
* learn more about [various software you need to create a website](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_software_do_I_need)
* move on to something practical like [how to upload files on a web server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Upload_files_to_a_web_server).

# What is the difference between webpage, website, web server, and search engine?

In this article, we describe various web-related concepts: *web pages, websites, web servers, and search engines*. These terms are often confused by newcomers to the web or are incorrectly used. Let's learn what they each mean!

|  |  |
| --- | --- |
| **Prerequisites:** | You should know [how the Internet works](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work). |
| **Objective:** | Be able to describe the differences between a web page, a website, a web server, and a search engine. |

## **Summary**

As with any area of knowledge, the web comes with a lot of jargon. Don't worry, we won't overwhelm you with all of it (we have a [glossary](https://developer.mozilla.org/en-US/docs/Glossary) if you're curious). However, there are a few basic terms you need to understand at the outset, since you'll hear these expressions all the time as you read on. It's easy to confuse these terms sometimes since they refer to related but different functionalities. In fact, you'll sometimes see these terms misused in news reports and elsewhere, so getting them mixed up is understandable!

We'll cover these terms and technologies in more detail as we explore further, but these quick definitions will be a great start for you:

**web page**

A document which can be displayed in a web browser such as Firefox, Google Chrome, Opera, Microsoft Internet Explorer or Edge, or Apple's Safari. These are also often called just "pages."

**website**

A collection of web pages which are grouped together and usually connected together in various ways. Often called a "web site" or simply a "site."

**web server**

A computer that hosts a website on the Internet.

**search engine**

A web service that helps you find other web pages, such as Google, Bing, Yahoo, or DuckDuckGo. Search engines are normally accessed through a web browser (e.g. you can perform search engine searches directly in the address bar of Firefox, Chrome, etc.) or through a web page (e.g. [bing.com](https://www.bing.com/) or [duckduckgo.com](https://duckduckgo.com/)).

Let's look at a simple analogy — a public library. This is what you would generally do when visiting a library:

1. Find a search index and look for the title of the book you want.
2. Make a note of the catalog number of the book.
3. Go to the particular section containing the book, find the right catalog number, and get the book.

Let's compare the library with a web server:

* The library is like a web server. It has several sections, which is similar to a web server hosting multiple websites.
* The different sections (science, math, history, etc.) in the library are like websites. Each section is like a unique website (two sections do not contain same books).
* The books in each section are like webpages. One website may have several webpages, e.g., the Science section (the website) will have books on heat, sound, thermodynamics, statics, etc. (the webpages). Webpages can each be found at a unique location (URL).
* The search index is like the search engine. Each book has its own unique location in the library (two books cannot be kept at the same place) which is specified by the catalog number.

## **Deeper dive**

So, let's dig deeper into how those four terms are related and why they are sometimes confused with each other.

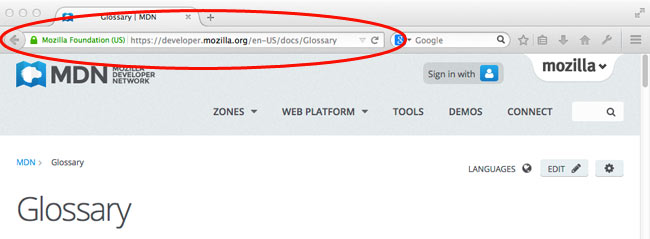
### **Web page**

A **web page** is a simple document displayable by a [browser](https://developer.mozilla.org/en-US/docs/Glossary/browser). Such documents are written in the [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML) language (which we look into in more detail in [other articles](https://developer.mozilla.org/en-US/docs/Web/HTML)). A web page can embed a variety of different types of resources such as:

* style information — controlling a page's look-and-feel
* scripts — which add interactivity to the page
* media — images, sounds, and videos.

**Note:**Browsers can also display other documents such as [PDF](https://developer.mozilla.org/en-US/docs/Glossary/PDF) files or images, but the term **web page** specifically refers to HTML documents. Otherwise, we only use the term **document**.

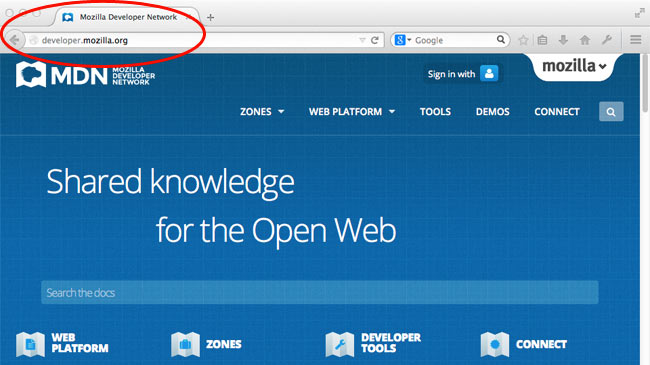
All web pages available on the web are reachable through a unique address. To access a page, just type its address in your browser address bar:



### **Web site**

A website is a collection of linked web pages (plus their associated resources) that share a unique domain name. Each web page of a given website provides explicit links—most of the time in the form of clickable portion of text—that allow the user to move from one page of the website to another.

To access a website, type its domain name in your browser address bar, and the browser will display the website's main web page, or homepage (casually referred as "the home"):



The ideas of a web page and a website are especially easy to confuse for a website that contains only one web page. Such a website is sometimes called a single-page website.

### **Web server**

A web server is a computer hosting one or more websites. "Hosting" means that all the web pages and their supporting files are available on that computer. The web server will send any web page from the website it is hosting to any user's browser, per user request.

Don't confuse websites and web servers. For example, if you hear someone say, "My website is not responding", it actually means that the web server is not responding and therefore the website is not available. More importantly, since a web server can host multiple websites, the term web server is never used to designate a website, as it could cause great confusion. In our previous example, if we said, "My web server is not responding", it means that no websites on that web server are available.

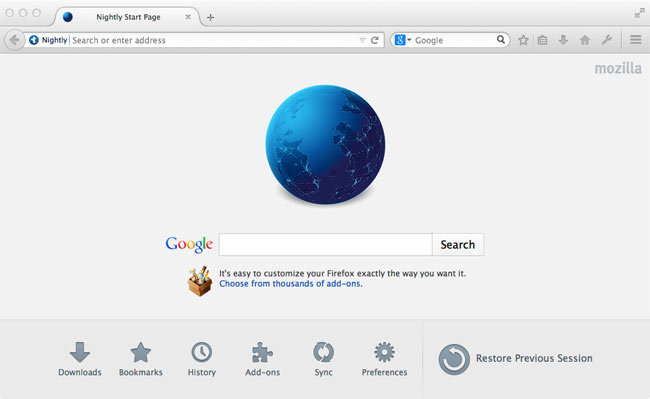
### **Search engine**

Search engines are a common source of confusion on the web. A search engine is a special kind of website that helps users find web pages from other websites.

There are plenty out there: [Google](https://www.google.com/), [Bing](https://www.bing.com/), [Yandex](https://www.yandex.com/), [DuckDuckGo](https://duckduckgo.com/), and many more. Some are generic, some are specialized about certain topics. Use whichever you prefer.

Many beginners on the web confuse search engines and browsers. Let's make it clear: A ***browser*** is a piece of software that retrieves and displays web pages; a **search engine** is a website that helps people find web pages from other websites. The confusion arises because, the first time someone launches a browser, the browser displays a search engine's homepage. This makes sense, because, obviously, the first thing you want to do with a browser is to find a web page to display. Don't confuse the infrastructure (e.g., the browser) with the service (e.g., the search engine). The distinction will help you quite a bit, but even some professionals speak loosely, so don't feel anxious about it.

Here is an instance of Firefox showing a Google search box as its default startup page:



## **Next steps**

* Dig deeper: [What is a web server](https://developer.mozilla.org/en-US/docs/Learn/What_is_a_web_server)
* See how web pages are linked into a web site: [Understanding links on the web](https://developer.mozilla.org/en-US/docs/Learn/Understanding_links_on_the_web)